```
PPPPPPPPPPPP    LLL              IIIIIIIII   RRRRRRRRRRRR    TTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPP    LLL              IIIIIIIII   RRRRRRRRRRRR    TTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPP    LLL              IIIIIIIII   RRRRRRRRRRRR    TTTTTTTTTTTTTTT  LLL
PPP        PPP  LLL                 III      RRR        RRR        TTT        LLL
PPP        PPP  LLL                 III      RRR        RRR        TTT        LLL
PPP        PPP  LLL                 III      RRR        RRR        TTT        LLL
PPP        PPP  LLL                 III      RRR        RRR        TTT        LLL
PPP        PPP  LLL                 III      RRR        RRR        TTT        LLL
PPPPPPPPPPPP    LLL                 III      RRRRRRRRRRRR          TTT        LLL
PPPPPPPPPPPP    LLL                 III      RRRRRRRRRRRR          TTT        LLL
PPPPPPPPPPPP    LLL                 III      RRRRRRRRRRRR          TTT        LLL
PPP             LLL                 III      RRR   RRR             TTT        LLL
PPP             LLL                 III      RRR   RRR             TTT        LLL
PPP             LLL                 III      RRR   RRR             TTT        LLL
PPP             LLL                 III      RRR      RRR          TTT        LLL
PPP             LLL                 III      RRR      RRR          TTT        LLL
PPP             LLL                 III      RRR      RRR          TTT        LLL
PPP             LLLLLLLLLLLLLLL  IIIIIIIII   RRR        RRR        TTT        LLLLLLLLLLLLLLL
PPP             LLLLLLLLLLLLLLL  IIIIIIIII   RRR        RRR        TTT        LLLLLLLLLLLLLLL
PPP             LLLLLLLLLLLLLLL  IIIIIIIII   RRR        RRR        TTT        LLLLLLLLLLLLLLL
```

```
_$2

Sym
---
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI

PLI
PLI
PLI

PLI
PLI
PLI
PLI
PLI
PLI
PLI

PLI
PLI
PLI
PLI

PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
```

```
**FILE**ID**PLIPUTLIS                              L 12




PPPPPPPP    LL            IIIIII    PPPPPPPP    UU      UU  TTTTTTTTTT  LL            IIIIII      SSSSSSSS
PPPPPPPP    LL            IIIIII    PPPPPPPP    UU      UU  TTTTTTTTTT  LL            IIIIII      SSSSSSSS
PP      PP  LL              II      PP      PP  UU      UU      TT      LL              II      SS
PP      PP  LL              II      PP      PP  UU      UU      TT      LL              II      SS
PP      PP  LL              II      PP      PP  UU      UU      TT      LL              II      SS
PP      PP  LL              II      PP      PP  UU      UU      TT      LL              II      SS
PPPPPPPP    LL              II      PPPPPPPP    UU      UU      TT      LL              II        SSSSSS
PPPPPPPP    LL              II      PPPPPPPP    UU      UU      TT      LL              II        SSSSSS
PP          LL              II      PP          UU      UU      TT      LL              II              SS
PP          LL              II      PP          UU      UU      TT      LL              II              SS
PP          LL              II      PP          UU      UU      TT      LL              II              SS      ....
PP          LL              II      PP          UU      UU      TT      LL              II              SS      ....
PP          LLLLLLLLLL    IIIIII    PP          UUUUUUUUUU      TT      LLLLLLLLLL    IIIIII    SSSSSSSS        ....
PP          LLLLLLLLLL    IIIIII    PP          UUUUUUUUUU      TT      LLLLLLLLLL    IIIIII    SSSSSSSS        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0000            .title  pli$putlistitem
0000            .ident  /1-002/                                 ; Edit WHM1002
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000
0000    ;++
0000    ; facility:
0000    ;
0000    ;       VAX/VMS PL1 runtime library
0000    ;
0000    ; abstract:
0000    ;
0000    ;       This module contains the pl1 runtime routines to put items to a
0000    ;       pl1 stream file under list directed i/o.
0000    ;
0000    ;
0000    ; author: c. spitz 28-nov-79
0000    ;
0000    ; modified:
0000    ;
0000    ;
0000    ;       1-002   Bill Matthews   29-September-1982
0000    ;
0000    ;               Invoke macros $defdat and rtshare instead of $defopr and share.
0000    ;
0000    ;--
0000
0000    ;
0000    ; external definitions
0000    ;
0000            $deffcb                         ;define file control block
0000            $defstk                         ;define stack frame offsets
0000            $defstr                         ;define stream block offsets
0000            $defdat                         ;define operand node data types
```

```
                            0000            $defgetopt                              ;define get options block
                            0000            $rabdef                                 ;define rms rab offsets
                            0000            $rmsdef                                 ;define rms error codes
                            0000            $ssdef                                  ;define system status codes
                            0000
                            0000    ;
                            0000    ; local data
                            0000    ;
                            0000
                            0000            rtshare                 ;sharable
                            0000
                            0000    ;++
                            0000    ; pli$putl****
                            0000    ;
                            0000    ; the pli$putl**** routines are called by the compiled code to put items
                            0000    ; to a stream output file under list directed transmission. each routine
                            0000    ; converts the source item to a character string based on the source data
                            0000    ; type, and puts then puts the string to the file by jumping to
                            0000    ; pli$$putnlis_r6.
                            0000    ;--
                            0000
                            0000
                            0000    ;pli$putlchar_r6
                            0000    ; inputs:
                            0000    ;       r0 - address of element to put
                            0000    ;       r1 - size/prec of element to put
                            0000    ;       r11 - address of stream block
                            0000    ;       ap - address of file control block
                            0000    ; outputs:
                            0000    ;       none
                            0000    ; side effects:
                            0000    ;       r0-r6 are destroyed
                            0000
                            0000    pli$putlchar_r6::
       0C AC   08   C8      0000            bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
          51   50   C0      0004            addl    r0,r1                   ;get ending address of source
    52    1A AB   9E        0007            movab   <str_b_field+2>(r11),r2 ;get starting addr in field
          54   52   D0      000B            movl    r2,r4                   ;copy it
    53    14 AB   D0        000E            movl    str_l_fld_end(r11),r3   ;get end addr of field
    03 0C AC   07   E0      0012            bbs     #atr_v_print,fcb_l_attr(ap),5$ ;skip lead quote if print
          82   27   90      0017            movb    #^x27,(r2)+             ;insert the leading quote
          51   50   D1      001A    5$:     cmpl    r0,r1                   ;nothing in source?
               29   13      001D            beql    60$                     ;if eql, then yes
          27   60   91      001F    10$:    cmpb    (r0),#^x27              ;next char a quote?
               0C   12      0022            bneq    20$                     ;if neq, then no
    07 0C AC   07   E0      0024            bbs     #atr_v_print,fcb_l_attr(ap),20$ ;if print file, don't change
     82    2727   8F B0     0029            movw    #^x2727,(r2)+           ;insert 2 quotes
               03   11      002E            brb     50$                     ;continue
          82   60   90      0030    20$:    movb    (r0),(r2)+              ;copy to field
       02 50   51   F2      0033    50$:    aoblss  r1,r0,55$               ;if not end of source, cont
               0F   11      0037            brb     60$                     ;finish it off
          53   52   D1      0039    55$:    cmpl    r2,r3                   ;field overflow?
               E1   19      003C            blss    10$                     ;if lss, then no, cont
 50    00000000'8F  D0      003E            movl    #pli$_strovfl,r0        ;set field overflow
             0113   31      0045            brw     fail                    ;and fail
    03 0C AC   07   E0      0048    60$:    bbs     #atr_v_print,fcb_l_attr(ap),70$ ;if print, don't add trail quote
          82   27   90      004D            movb    #^x27,(r2)+             ;insert trailing quote
```

```
              52   54   C2  0050   70$:    subl    r4,r2                       ;get length
              74   52   B0  0053           movw    r2,-(r4)                    ;set length in field
           00000000'GF  16  0056           jsb     g^pli$$putnlis_r6           ;put in buffer
           0C AC   08   CA  005C           bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                        05  0060           rsb                                 ;return
                            0061
                            0061   ;pli$putlvcha_r6
                            0061   ;       inputs:
                            0061   ;           r0 - address of element to put
                            0061   ;           r1 - size/prec of element to put
                            0061   ;           r11 - address of stream block
                            0061   ;           ap - address of file control block
                            0061   ;       outputs:
                            0061   ;           none
                            0061   ;       side effects:
                            0061   ;           r0-r6 are destroyed
                            0061   ;
                            0061
                            0061   pli$putlvcha_r6::
           0C AC   08   C8  0061           bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
              51   80   3C  0065           movzwl  (r0)+,r1                    ;get length of source
                 FF95   31  0068           brw     pli$putlchar_r6             ;continue in common
                            006B
                            006B
                            006B
                            006B   ;pli$putlbit_r6
                            006B   ;       inputs:
                            006B   ;           r0 - address of element to put
                            006B   ;           r1 - size/prec of element to put
                            006B   ;           r2 - offset to starting bit
                            006B   ;           r11 - address of stream block
                            006B   ;           ap - address of file control block
                            006B   ;       outputs:
                            006B   ;           none
                            006B   ;       side effects:
                            006B   ;           r0-r6 are destroyed
                            006B   ;
                            006B
                            006B   pli$putlbit_r6::
           0C AC   08   C8  006B           bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
              55   52   D0  006F           movl    r2,r5                       ;copy offset
          52   18  AB   9E  0072           movab   str_b_field(r11),r2         ;get field addr
      82   03      51   A1  0076           addw3   r1,#3,(r2)+                 ;set size
              82   27   90  007A           movb    #^x27,(r2)+                 ;insert a quote
              53   51   D0  007D           movl    r1,r3                       ;get size
    000003E8 8F    53   D1  0080           cmpl    r3,#1000                    ;field overflow?
                   0A   15  0087           bleq    10$                         ;if leq, then no
      50   00000000'8F  D0  0089           movl    #pli$_strovfl,r0            ;set field over flow
                 00C8   31  0090           brw     fail                        ;and fail
      54   52      51   C1  0093   10$:    addl3   r1,r2,r4                    ;get addr of end of string
      64   4227    8F   B0  0097           movw    #^x4227,(r4)                ;plug in trailing quote and B
  00000000'GF      00   FB  009C           calls   #0,g^pli$bitchar_r6         ;convert bits
           00000000'GF  16  00A3           jsb     g^pli$$putnlis_r6           ;put in buffer
           0C AC   08   CA  00A9           bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                        05  00AD           rsb                                 ;return
                            00AE
                            00AE
```

```
                             00AE  ;pli$putlabit_r6
                             00AE  ;     inputs:
                             00AE  ;        r0 - address of element to put
                             00AE  ;        r1 - size/prec of element to put
                             00AE  ;        r11 - address of stream block
                             00AE  ;        ap - address of file control block
                             00AE  ;     outputs:
                             00AE  ;        none
                             00AE  ;     side effects:
                             00AE  ;        r0-r6 are destroyed
                             00AE  ;
                             00AE
                             00AE  pli$putlabit_r6::
           0C AC    08   C8  00AE        bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
                    52   D4  00B2        clrl    r2                       ;set offset to 0
                    B5   11  00B4        brb     pli$putlbit_r6           ;join common code
                             00B6
                             00B6  ;pli$putlfixb_r6
                             00B6  ;     inputs:
                             00B6  ;        r0 - address of element to put
                             00B6  ;        r1 - size/prec of element to put
                             00B6  ;        r11 - address of stream block
                             00B6  ;        ap - address of file control block
                             00B6  ;     outputs:
                             00B6  ;        none
                             00B6  ;     side effects:
                             00B6  ;        r0-r6 are destroyed
                             00B6  ;
                             00B6
                             00B6  pli$putlfixb_r6::
           0C AC    08   C8  00B6        bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
            52   18 AB   9E  00BA        movab   str_b_field(r11),r2      ;set field addr
      53   000003E8 8F   D0  00BE        movl    #1000,r3                 ;set size
    00000000'GF     00   FB  00C5        calls   #0,g^pli$fixbvcha_r6     ;convert it
       00000000'GF  16       00CC        jsb     g^pli$$putnlis_r6        ;put in buffer
           0C AC    08   CA  00D2        bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                         05  00D6        rsb                              ;return
                             00D7
                             00D7
                             00D7  ;pli$putlfixd_r6
                             00D7  ;     inputs:
                             00D7  ;        r0 - address of element to put
                             00D7  ;        r1 - size/prec of element to put
                             00D7  ;        r11 - address of stream block
                             00D7  ;        ap - address of file control block
                             00D7  ;     outputs:
                             00D7  ;        none
                             00D7  ;     side effects:
                             00D7  ;        r0-r6 are destroyed
                             00D7  ;
                             00D7
                             00D7  pli$putlfixd_r6::
           0C AC    08   C8  00D7        bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
            52   18 AB   9E  00DB        movab   str_b_field(r11),r2      ;set field addr
      53   000003E8 8F   D0  00DF        movl    #1000,r3                 ;set size
    00000000'GF     00   FB  00E6        calls   #0,g^pli$fixdvcha_r6     ;convert it
       00000000'GF  16       00ED        jsb     g^pli$$putnlis_r6        ;put in buffer
```

```
          0C AC    08    CA  00F3              bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                         05  00F7              rsb                                 ;return
                             00F8
                             00F8
                             00F8    ;pli$putlfltb_r6
                             00F8    ;      inputs:
                             00F8    ;          r0 - address of element to put
                             00F8    ;          r1 - size/prec of element to put
                             00F8    ;          r11 - address of stream block
                             00F8    ;          ap - address of file control block
                             00F8    ;      outputs:
                             00F8    ;          none
                             00F8    ;      side effects:
                             00F8    ;          r0-r6 are destroyed
                             00F8    ;
                             00F8
                             00F8    pli$putlfltb_r6::
          0C AC    08    C8  00F8              bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
          52    18 AB    9E  00FC              movab   str_b_field(r11),r2     ;set field addr
    53   000003E8 8F     D0  0100              movl    #1000,r3                ;set field width
   00000000'GF     00    FB  0107              calls   #0,g^pli$fltbvcha_r6    ;convert it
      00000000'GF        16  010E              jsb     g^pli$$putnlis_r6       ;put in buffer
          0C AC    08    CA  0114              bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                         05  0118              rsb                             ;return
                             0119
                             0119
                             0119    ;pli$putlfltd_r6
                             0119    ;      inputs:
                             0119    ;          r0 - address of element to put
                             0119    ;          r1 - size/prec of element to put
                             0119    ;          r11 - address of stream block
                             0119    ;          ap - address of file control block
                             0119    ;      outputs:
                             0119    ;          none
                             0119    ;      side effects:
                             0119    ;          r0-r6 are destroyed
                             0119    ;
                             0119
                             0119    pli$putlfltd_r6::
          0C AC    08    C8  0119              bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
          52    18 AB    9E  011D              movab   str_b_field(r11),r2     ;set field addr
    53   000003E8 8F     D0  0121              movl    #1000,r3                ;set field width
   00000000'GF     00    FB  0128              calls   #0,g^pli$fltdvcha_r6    ;convert it
      00000000'GF        16  012F              jsb     g^pli$$putnlis_r6       ;put in buffer
          0C AC    08    CA  0135              bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                         05  0139              rsb                             ;return
                             013A
                             013A
                             013A    ;pli$putlpic_r6
                             013A    ;      inputs:
                             013A    ;          r0 - address of element to put
                             013A    ;          r1 - size/prec of element to put
                             013A    ;          r11 - address of stream block
                             013A    ;          ap - address of file control block
                             013A    ;      outputs:
                             013A    ;          none
                             013A    ;      side effects:
                                013A   ;       r0-r6 are destroyed
                                013A
                                013A   pli$putlpic_r6::
        0C AC    08      C8     013A           bisl    #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
        52    18 AB      9E     013E           movab   str_b_field(r11),r2      ;set field addr
 53   000003E8 8F        D0     0142           movl    #1000,r3                 ;set size
 00000000'GF     00      FB     0149           calls   #0,g^pli$picvcha_r6      ;convert it
      00000000'GF        16     0150           jsb     g^pli$$putnlis_r6        ;put in buffer
        0C AC    08      CA     0156           bicl    #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
                         05     015A           rsb                              ;return
                                015B
        08 AC    50      D0     015B   fail:   movl    r0,fcb_l_error(ap)       ;set error in fcb
                         5C     DD     015F    pushl   ap                       ;set fcb address
                         50     DD     0161    pushl   r0                       ;set error code
      00000000'8F        DD     0163           pushl   #pli$_error              ;set error condition
 00000000'GF     03      FB     0169           calls   #3,g^pli$io_error        ;signal error condition
                         04     0170           ret                              ;return
                                0171
                                0171           .end
```

```
                                                        F 13
PLI$PUTLISTITEM                                                   16-SEP-1984 02:24:21  VAX/VMS Macro V04-00          Page   7
Symbol table                                                       6-SEP-1984 11:39:33  [PLIRTL.SRC]PLIPUTLIS.MAR;1         (1)

ATR_M_RECUR                      = 00000008             PLI$_STROVFL                ********   X   02
ATR_V_PRINT                      = 00000007             SIZ...                    = 00000001
FAIL                               0000015B R      02   STK_L_AP                    00000008
FCB_B_ENVIR                        000001C2             STK_L_ARG_LIST              FFFFFFF8
FCB_B_ESA                          0000012E             STK_L_CND_HND               00000000
FCB_B_EXTRA                        0000003D             STK_L_CND_LST               FFFFFFF4
FCB_B_FAB                          000000A6             STK_L_DISPLAY               FFFFFFFC
FCB_B_IDENT                        00000040             STK_L_FP                    0000000C
FCB_B_IDENT_NAM                    00000042             STK_L_PC                    00000010
FCB_B_NAM                          000000F6             STK_L_PSL                   00000004
FCB_B_NUMKCBS                      0000003C             STK_L_REGS                  00000014
FCB_B_RAB                          00000062             STR_B_FIELD                 00000018
FCB_C_LEN                          000001C2             STR_C_LEN                   00000C08
FCB_C_STRLEN                       00000034             STR_L_FLD_END               00000014
FCB_L_ATTR                         0000000C             STR_L_FLD_PT                00000010
FCB_L_BUF                          00000014             STR_L_FP                    00000004
FCB_L_BUF_END                      00000018             STR_L_FS                    0000000C
FCB_L_BUF_PT                       0000001C             STR_L_PARENT                00000008
FCB_L_CNDADDR                      000001B2             STR_L_SP                    00000000
FCB_L_CONDIT                       000001AE             STR_L_STACK                 00000C04
FCB_L_DTTR                         00000010             STR_L_STACK_END             00000408
FCB_L_ERROR                        00000008
FCB_L_KCB                          00000038
FCB_L_NEXT                         00000000
FCB_L_PREVIOUS                     00000004
FCB_L_PRN                          00000034
FCB_Q_RFA                          00000020
FCB_W_COLUMN                       0000002E
FCB_W_IDENT_LEN                    00000040
FCB_W_LINE                         00000030
FCB_W_LINESIZE                     0000002A
FCB_W_PAGE                         00000032
FCB_W_PAGESIZE                     0000002C
FCB_W_REVISION                     00000028
GETOPT_B_BITS                      00000009
GETOPT_B_TMO                       00000008
GETOPT_C_LEN                       0000000A
GETOPT_L_FXDCTL                    00000000
GETOPT_L_PROMPT                    00000004
PLI$$PUTRLIS_R6                    ********    X   02
PLI$BITCHAR_R6                     ********    X   02
PLI$FIXBVCHA_R6                    ********    X   02
PLI$FIXDVCHA_R6                    ********    X   02
PLI$FLTBVCHA_R6                    ********    X   02
PLI$FLTDVCHA_R6                    ********    X   02
PLI$IO_ERROR                       ********    X   02
PLI$PICVCHA_R6                     ********    X   02
PLI$PUTLABIT_R6                    000000AE RG     02
PLI$PUTLBIT_R6                     0000006B RG     02
PLI$PUTLCHAR_R6                    00000000 RG     02
PLI$PUTLFIXB_R6                    000000B6 RG     02
PLI$PUTLFIXD_R6                    000000D7 RG     02
PLI$PUTLFLTB_R6                    000000F8 RG     02
PLI$PUTLFLTD_R6                    00000119 RG     02
PLI$PUTLPIC_R6                     0000013A RG     02
PLI$PUTLVCHA_R6                    00000061 RG     02
PLI$_ERROR                         ********    X   02
```

```
                                                +-----------------+
                                                ! Psect synopsis !
                                                +-----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (  0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                           FFFFFFFC  (     0.)  01 (  1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
_PLI$CODE                       00000171  (   369.)  02 (  2.)    PIC    USR    CON    REL    LCL   SHR   EXE   RD   NOWRT NOVEC LONG

                                            +------------------------+
                                            ! Performance indicators !
                                            +------------------------+

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                        12    00:00:00.08    00:00:00.29
Command processing                    74    00:00:00.62    00:00:01.95
Pass 1                               251    00:00:09.47    00:00:19.42
Symbol table sort                      0    00:00:01.34    00:00:02.66
Pass 2                                59    00:00:01.62    00:00:03.81
Symbol table output                   10    00:00:00.08    00:00:00.28
Psect synopsis output                  1    00:00:00.03    00:00:00.23
Cross-reference output                 0    00:00:00.00    00:00:00.00
Assembler run totals                 407    00:00:13.24    00:00:28.65

The working set limit was 1050 pages.
51576 bytes (101 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 972 non-local and 8 local symbols.
304 source lines were read in Pass 1, producing 11 object records in Pass 2.
18 pages of virtual memory were used to define 16 macros.

                                           +----------------------------+
                                           ! Macro library statistics !
                                           +----------------------------+

Macro library name                         Macros defined
------------------                         --------------
_$255$DUA28:[PLIRTL.OBJ]PLIRTMAC.MLB;1                  6
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       7
TOTALS (all libraries)                                 13

995 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=TRACEBACK/LIS=LIS$:PLIPUTLIS/OBJ=OBJ$:PLIPUTLIS MSRC$:PLIPUTLIS/UPDATE=(ENH$:PLIPUTLIS)+LIB$:PLIRTM
```

0308 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PLIFORMAT
LIS

PLIGETBUF
LIS

PLIMSGTXT
LIS

PLIGETEDI
LIS

PLIHEEP
LIS

PLIPUTFIL
LIS

PLIRMSBIS
LIS

PLIRECOPT
LIS

PLIOPEN
LIS

PLIREAD
LIS

PLIPROTEC
LIS

PLIREWRIT
LIS

PLIGETLIS
LIS

PLIPUTEDI
LIS

PLIPKDIUL
LIS

PLIPUTLIS
LIS

PLIMSGPTR
LIS

PLIPKDIUS
LIS

PLIPUTBUF
LIS

PLIGETFIL
LIS